Trabajo Guiado MOA

Juanjo Sierra

25 de abril de 2019

Ejercicio 1

Se pide comparar la eficacia de un Hoeffding Tree con un clasificador Naïve Bayes, para un flujo de datos de 1.000.000 de instancias generadas con un generador RandomTreeGenerator, suponiendo una frecuencia de muestreo de 10.000 y con el método de evaluación Interleaved Test-Then-Train.

Aquí vemos como se ejecuta un modelo con Naïve Bayes.

```
java -cp moa.jar -javaagent:sizeofag.jar moa.DoTask "EvaluateInterleavedTestThenTrain \
-l bayes.NaiveBayes -s generators.RandomTreeGenerator -i 1000000 -f 10000"
```

Y aquí el ejemplo análogo con Hoeffding Tree.

```
java -cp moa.jar -javaagent:sizeofag.jar moa.DoTask "EvaluateInterleavedTestThenTrain \
-l trees.HoeffdingTree -s generators.RandomTreeGenerator -i 1000000 -f 10000"
```

Hay que generar una población de resultados, ejecutando cada uno de estos ejemplos anteriores 30 veces con 30 semillas distintas, y luego mediante un test comparar si existen diferencias significativas en los resultados obtenidos por los dos algoritmos. La generación de las poblaciones de resultados se puede conseguir mediante scripts.

Una vez obtenidos los datos, cada archivo convenientenemente renombrado acorde al algoritmo que lo ha generado, los podemos leer con R.

```
poblacionNaiveBayes = array(dim = 30)
for (i in 1:30) {
  archivo =
    paste(c(paste(c("./Datos/Ejercicio1/naiveBayes",i),collapse = ""),".csv"),collapse="")
  datos = read.csv(archivo)
  accuracyFinal =
    datos$classifications.correct..percent.[length(datos$classifications.correct..percent.)]
  poblacionNaiveBayes[i] = accuracyFinal
}
```

Y a continuación leemos los de Hoeffding.

```
poblacionHoeffding = array(dim = 30)
for (i in 1:30) {
  archivo =
    paste(c(paste(c("./Datos/Ejercicio1/hoeffding",i),collapse = ""),".csv"),collapse="")
  datos = read.csv(archivo)
  accuracyFinal =
    datos$classifications.correct..percent.[length(datos$classifications.correct..percent.)]
  poblacionHoeffding[i] = accuracyFinal
}
```

Para saber si hacer un test paramétrico o no paramétrico para comparar los resultados de ambos algoritmos, primero evaluamos con el test de Shapiro-Wilk si siguen una distribución normal ambas poblaciones.

```r
shapiro.test(poblacionNaiveBayes)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  poblacionNaiveBayes
## W = 0.97381, p-value = 0.6478
```

```r
shapiro.test(poblacionHoeffding)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  poblacionHoeffding
## W = 0.9824, p-value = 0.8852
```

Como según los resultados las dos poblaciones siguen una distribución normal podemos realizar un test paramétrico como el test T de Student.

```r
t.test(poblacionNaiveBayes, poblacionHoeffding)
```

```
##
##  Welch Two Sample t-test
##
## data:  poblacionNaiveBayes and poblacionHoeffding
## t = -1055.4, df = 45.572, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -20.91581 -20.83615
## sample estimates:
## mean of x mean of y
##  73.67307  94.54905
```

En base a los resultados obtenidos podemos afirmar que la diferencia en media de los resultados de los dos algoritmos no es igual a 0, es decir, existen diferencias significativas. En este caso, el algoritmo Hoeffding funciona mejor que el algoritmo Naïve Bayes, pues a igualdad de distribución tiene una media mayor.